

Tekla Structures sovellus- sähkösuunnitelman liittäminen elementtipiirustuksen

Remus Kaczmarek



Tekijä Remus Kaczmarek	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Tekla Structures sovellus- sähkösuunnitelman liittäminen elementtipiirustuksen	Sivu- ja liitesivumäärä 29 + 0
Opinnäytetyön otsikko englanniksi Application of Tekla Structures- adding an electricity design to the drawing element	
<p>Työn tavoitteena oli luoda sovellus, jonka avulla pystytään automatisoida Tekla Structures -rakennesuunnitteluohjelmiston perustoiminto. Sovelluksen tarkoituksena on automatisoida sähköistyksen lisääminen elementtipiirustukseen. Sovelluksen kehittämisessä käytettiin Tekla Open API – ohjelmistorajapintaa. Kehityksessä aihe rajattiin sovelluksen luomiseen ja testaus sekä käyttöönotto jätettiin pois työstä.. Opinnäytetyön toimeksiantajana toimi Sweco Rakenetekniikka Oy.</p> <p>Opinnäytetyön teoriaosuudessa esitellään Tekla Structuresin, Tekla Open API:n ja elementtisuunnittelun perusteet. Lisäksi selitetään sovelluksen kehitysvaiheita ja sen toiminnallisuus. Opinnäytetyön lähteinä käytettiin Teklalta saatavia dokumentteja, haastatteluja asiantuntijoiden kanssa sekä internetistä löydettäviä asiakirjoja.</p> <p>Sovelluksen kehittämisessä käytettiin Microsoft Visual Studio 2015 kehitysympäristöä ja sovellusten ohjelmointikielenä oli C#. Sovelluksen kehittäminen Tekla Structures-ohjelmistoon oli mahdollista Teklan oman avoimen Tekla Open API- ohjelmointirajapinnan avulla, joka toimii Microsoftin .NET- alustalla. Sovelluksen käyttöliittymä toteutettiin Console Application-tyyppisenä.</p> <p>Työ aloitettiin tammikuun 2016 puolivälissä ja se kesti noin kolme kuukautta. Projektin kehitysvaiheen tuloksena saatiin Tekla Structures-ohjelmistoon toimiva ja helppokäyttöinen sovellus, joka automatisoi sähköistyksen lisäämisen elementtipiirustukseen. Sovelluksen ansiosta Tekla Structures -käyttäjät pystyvät nopeuttamaan työskentelyään ja välttämään tarpeettomia virheitä. Toimivan sovellusotetaan yrityksessä myöhemmin käyttöön.</p>	
Asiasanat Sovellus, Tekla Structures, API, C#, sähkösuunnitelma, elementtipiirustus	

Author Remus Kaczmarek	
Degree programme Business Information Technology	
Report/thesis title Application of Tekla Structures- adding an electricity design to the drawing element	Number of pages and appendix pages 29 + 0
<p>The aim was to create an application that allows to automate a basic function in the Tekla Structures - the basic structural design of the software. The application is designed to automate the addition of electrification to the element drawing. During the process of application development Tekla Open API application - software interface was used. The application development was limited to the creation of the application. Testing and deployment were left outside the scope of this study. This thesis was commissioned by Sweco Structural Engineering Oy.</p> <p>The theoretical part of the thesis presents the backgrounds to Tekla Structures software, Tekla Open API method and element designs. In addition, it was explained the development procedure steps of the application and its functionality. Sources used in the thesis are derived from Tekla documents, the interviews with experts and the documents found on the Internet.</p> <p>In the development of the application, "Microsoft Visual Studio 2015" application development environment and programming language, C # were used. With the help of Tekla's structures software Application Development, it is possible to open your Tekla Open API using an API that runs on Microsoft, NET platform. The application user interface was implement as Console Application type.</p> <p>The work was initiated in mid-January 2016 and it took about three months. The project resulted from the development phase to Tekla Structures software functional and easy-to-use application that automates the process of adding electrification to the element drawing. The application allows Tekla Structures users to accelerate their work and to avoid unnecessary mistakes. The development of the functioning application has been very successful and it will be implemented later for the company's use as well.</p>	
Keywords Application, Tekla Structures, API, C#, electricity design, element drawing	

Sisällys

1	Johdanto	1
1.1	Työn tavoite ja rajausta	1
1.2	Käsitteet	2
2	Tekla Structures	3
2.1	Ohjelmiston esittely	3
2.2	Hyötyjä käytöstä	4
2.3	Tekla Structures Full-ohjelmistokokoonpano	5
3	Tekla Open API	8
3.1	Yleistä	8
3.2	Tekla Open API- rajapinnan osat	9
3.3	Sovellukset ja plugins	11
4	Sähkösuunnitelmat osaksi elementtisuunnitelmia	12
4.1	Rakennussuunnittelu	12
4.2	Elementtisuunnittelu	12
4.3	Sähköistysten lisääminen elementteihin	15
5	Sovellus sähkösuunnitelman liittämiseksi elementtipiirustukseen	17
5.1	Sovelluksen tarve	17
5.2	Prosessi	17
5.3	Sovelluksen kehittäminen	18
5.4	Piirustusten avaaminen ja käsittely	20
5.5	Piirustusten vienti DWG-tiedostona	24
5.6	Sähköpiirustuspohjan lisääminen DWG- referenssinä	25
6	Yhteenveto	28

1 Johdanto

Rakennushanke on monivaiheinen prosessi, joka on suunniteltava alusta loppuun ennen toteutusta. Rakennuksen suunnittelu on vaativaa ja aikaa vievää. Tässä opinnäytetyössä luodaan Tekla Structures sovellus, joka helpottaa elementtipiirustuksien tuotantoa. Aihe on valittu, koska työharjoittelun toimeksiantajani Sweco Finland Oy ehdotti uuden sovelluksen toteuttamisesta Swecon Tekla Structures -ohjelmistoa käyttäviä työntekijöitä varten.

Valitsin projektin, koska olen kiinnostunut ohjelmoinnista ja minulla on kokemuksia rakennustekniikan alalta, mikä helpottaa tekemistä sovelluksen käyttötarkoituksen ymmärtämistä. Lisäksi halusin oppia enemmän ohjelmointia, jotta voisin hyödyntää kyseisiä taitoja tulevaisuudessa omassa työpaikassani.

Tekla Structures on ohjelmisto, jolla avulla pystytään toteuttamaan 3D-tietomallinnus rakennuksista. Tietomalli sisältää rakennuksen olennaiset rakentamiseen tarvittavat tiedot, kuten rakenneosat ja niiden väliset liitokset. Ohjelmistoon voidaan luoda sovelluksia, joita pystytään käyttämään Tekla Open -rajapinnan avulla. Opinnäytetyössä luotu sovellus liittyy tilanteeseen, jossa elementtien sähkömerkinnät tehdään elementtipiirustukseen sähkösuunnittelijan toimesta. Valmis sovellus tullaan lataamaan ja käyttämään Tekla Structures ympäristössä.

Opinnäytetyössä tutustutaan aluksi Tekla Structures -ohjelman käyttötarkoitukseen ja sen toimintoihin. Seuraavaksi selitetään Tekla Open API -ohjelman yleiskäsite ja sen merkitys makron luomisessa. Opinnäytetyön viimeinen kappale on projektityyppinen, siinä esitellään tarkasti jokainen työvaihe, tarvittavien ohjelmien asennus ja sovelluksen tekeminen.

Opinnäytetyöni tutkimuskysymykset ovat:

1. Miten Tekla Structures ohjelmaan luodaan sovellus?
2. Miten Tekla Open API:a käytetään sovelluksen luomisessa?
3. Miten sähkösuunnitelmien liittäminen elementtipiirustuksiin automatisoidaan?

1.1 Työn tavoite ja rajaus

Projektin tavoitteena on luoda mahdollisimman helppokäyttöinen sovellus, jolla avulla pystytään automatisoimaan Tekla Structures -ohjelman perustoiminto. Sovelluksen ansiosta Tekla Structures -käyttäjät pystyvät nopeuttamaan työskentelyään ja välttämään tarpeettomia virheitä. Tarkoituksena on yksinkertaistaa ja automatisoida piirustuksien tuotantoa,

mikä yrityksen kannalta säästää aikaa ja kuluja. Elementtipiirustuksen luominen koostuu monesta vaiheesta, ja kehitysovelluksen avulla automatisoidaan sähköistyksen lisääminen elementtipiirustukseen. Sovelluksen toteuttamiseen ei kuulu testaaminen eikä käyttöönotto. Niiden suorittamiseen tulee varata paljon enemmän aikaa, eikä sitä voida toteuttaa tässä oppinäytetyössä. Sovelluksen toimivuus tarkistetaan koodaamisen aikana, mutta varsinainen käyttöönotto ja testaus toteutetaan projektin jälkeen ja se valvotaan yrityksen sisällä.

1.2 Käsitteet

API	Application Programming Interface, eli ohjelmointirajapinta
BIM	Building Information Model eli rakennuksen tietomalli
CAD	Computer Aided Design eli tietokone avusteinen suunnittelu
COM	Component Object Model eli binaarinen liitäntä standardi ohjelmistokomponenteille
DWG	CAD – ohjelmien käyttämä tiedostomuoto
IFC	Industry Foundation Classes eli kansainvälinen ja jatkuvasti kehitettävä rakennusalan standardi oliopohjaisen tiedon siirtoon tietokonejärjestelmästä toiseen
UI	User Interface eli käyttöliittymä on se laitteen, ohjelmiston tai minkä tahansa muun tuotteen osa, jonka kautta käyttäjä käyttää tuotetta
.cs	Lähdekooditiedoston tyyppi, kirjoitettu C#:ssä
.dll	Dynamic-link Library, eli Microsoftin toteutus jaetun kirjaston käsiteestä Microsoft Windows:ssä ja OS / 2 käyttöjärjestelmissä

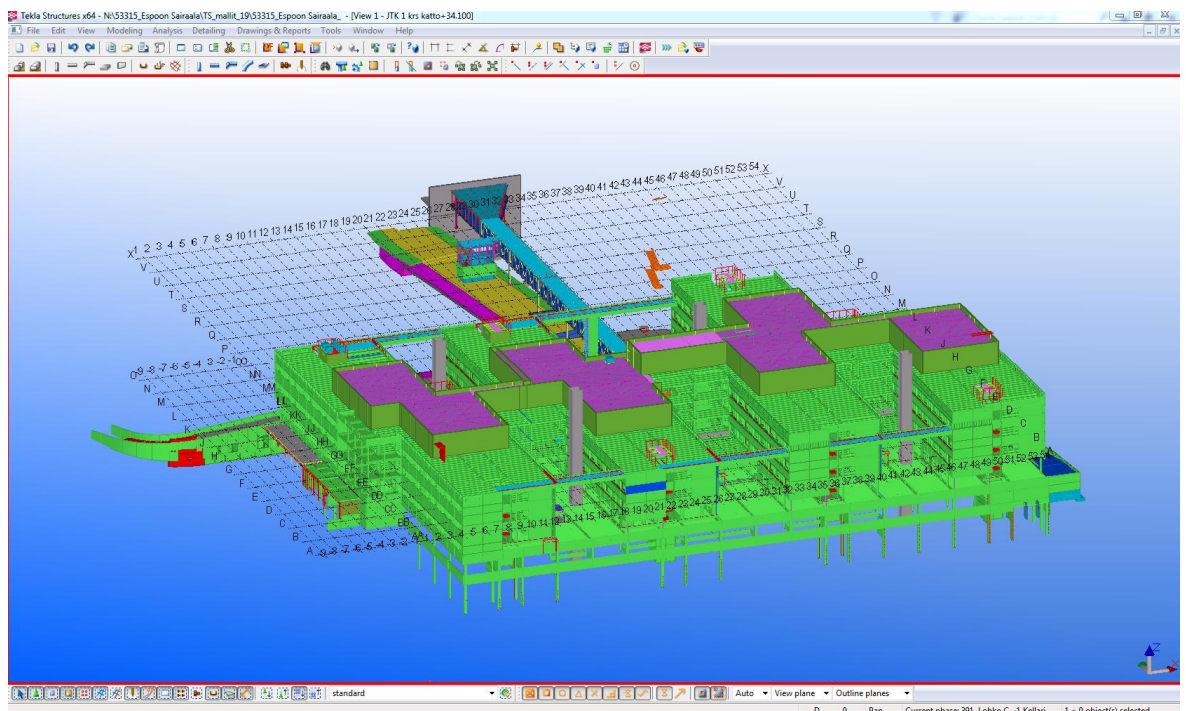
2 Tekla Structures

Luvussa esitellään Tekla Structures ohjelmisto ja sen käyttötarkoitus, kerrotaan Tekla Oyj -yrityksestä lyhyesti ja esitellään ohjelmiston käytön hyötyjä hankkeen toteuttamisessa kaikille osapuolille. Ohjelmiston toiminnallisuus kuvataan Tekla Structures Full -ohjelmistokokoonpanon avulla, jotta voidaan selventää Tekla ominaisuuksia ja sen käyttöä.

2.1 Ohjelmiston esittely

Tekla Structures on Tekla Oyj:n tuote. Suomalainen yritys on perustettu vuonna 1966 ja on kuulunut Trimble-konserniin vuodesta 2011 alkaen. Vuonna 2016 alusta Tekla Oyj on sulautunut Trimble-konserniin. Pääkonttori sijaitsee Espoossa ja maailmassa on yli 12 toimipistettä. Tekla on luonut yhteyttä asiakkaiden kanssa lähes 100 maassa. Vuonna 2010 on myyty yli 18 000 Tekla Structures-lisenssiä maailmanlaajuisesti. (Tekla 2016a.)

Tekla Structures on ohjelmisto, jonka avulla pystytään toteuttamaan 3D-tietomallinnusta. Tekla Structures on suunnattu pääosin rakennesuunnitteluun, mutta sitä myös voidaan käyttää rakentamisen hallintaan ja laskentaohjelmistojen kanssa. Kuvassa 1 on esitelty esimerkki Espoon sairaalan tietomallista. Projekti toteutettiin vuonna 2013–2015 ja rakennus valmistuu vuonna 2016. (Tekla 2016c.)



Kuva 1 Tekla Structures – tietomalli

2.2 Hyötyjä käytöstä

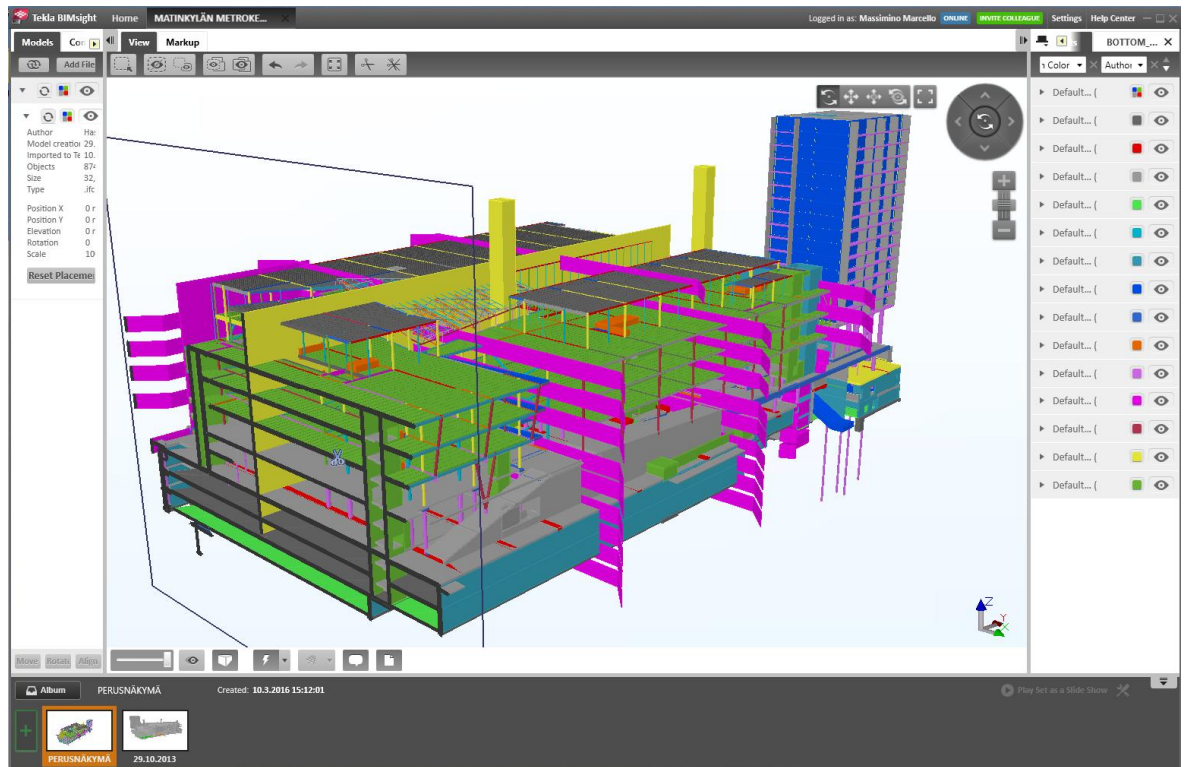
Ohjelmiston avulla pystytään mallintamaan erilaisia rakenteita. Mallintamiseen voidaan käyttää mitä tahansa materiaalia materiaaliluettelosta, kuten betonia, puuta, terästä tai muita materiaaleja. Rakenteet voi mallintaa käsin tai käyttää valmiita liitoksia ja detaljeja, jotka sijaitsevat Tekla Structuresin komponenttikirjastossa. Tekla Structuresissa voidaan myös luoda omia komponenttia tai makroja, mikä tapahtuu Tekla Open API-ohjelmointirajapinnan avulla. Lisäksi Tekla Structures voidaan yhdistää eri laskentaohjelmistoihin ja tietomallista voi ajaa monenlaisia raportteja esim. Microsoft Excelliin. (Tekla 2016b.)

Tekla Structures mahdollistaa osapuolten välisen tiedon jakamisen. Ohjelmiston avulla tietoja välitetään sujuvasti suunnittelijoiden, arkkitehtien ja urakoitsijoiden välillä. Tekla Structures sisältää multiuser-ominaisuuden, jonka avulla monet käyttäjät voivat muokata ja tarkastella ajantasaista tietomallia samanaikaisesti. Piirtäjien ja suunnittelijoiden työskentely yhtäaikaista samassa mallissa helpottaa tietomallin päivittämistä ja ylläpitoa. Lisäksi Tekla structures -mallista voi julkaista tietomalleja standardimuodoissa, kuten IFC ja DWG. IFC on tallennusformaatti, jolla eri suunnitteluohjelmilla tehtyjä tiedostoja voidaan lukea ja DWG on CAD – ohjelmien käyttämä tiedostomuoto. Suunnittelualojen tietomallit sovitetään yhteen julkaistujen standardimallien avulla. (Tekla 2016b.)

Tekla Structures -mallin avulla voidaan tutustua hankkeeseen ennen rakentamista ja hahmottaa projektin kokonaisuutta. Mallin visuaalinen tarkastelu helpottaa työskentelyä, projektin suunnittelua ja projektin hallintaa, sillä siitä pystytään helposti tunnistaa mahdolliset ongelmat ja saada ratkaisu ennen rakennustyömaan alkamista. Tämä toiminto mahdollistaa kalliiden viivästyksien ja lisä- ja muutostöiden välttämisen. Tekla Structures -työkalut auttavat mittaamaan ja hinnoittelemaan kaikki materiaalit, resurssit ja prosessit, mitä rakennushankkeeseen sisältyy. Niiden avulla projektissa voidaan laatia kustannuslaskenta ja aikataulu, joiden avulla projektit toteutetaan ajoissa ja budjetin rajoissa. Myös työmaalla Tekla Structuresilla on todella tärkeä rooli. Ohjelma helpottaa materiaalitilauksia ja logistiikkaa sekä päivittäisten ongelmien ratkaisua, jotka johtuvat esimerkiksi muutoksenhallinnasta. Työmaanhenkilöstöllä on mahdollisuus tarkastella mallia päivittäin, joten he pystyvät ymmärtämään paremmin projektin suunnitelmia ja tehtäviä. Tekla Structuresin käyttö työmaalla mahdollistaa myös suunnitteluvirheiden havaitsemisen ajoissa ja tiedon välityksen suunnittelijoille. (Tekla 2016c.)

Tekla Structures on yhteensopiva muiden ohjelmistojen kanssa, kuten AutoCAD:in tai Solibrin. Käyttäjä voi tuoda malliin tai viedä mallista eri tiedostoja, sillä Tekla Structures

tukee eri tiedonsiirtoformaatteja, kuten DWG-muotoa. Asiakas voi avata ja tarkastella tietomallista tehtyä IFC-muodossa projektipankkiin tallennettua mallia esim. ilmaisella Tekla BIMsight -ohjelmalla, joka esitellään kuvassa 2. Tekla BIMsight:in käyttöön ei tarvita Tekla Structures -lisenssiä. Tekla Structuresin Viewer-lisenssin avulla voidaan pelkästään tarkastella mallia tekemättä siihen muutoksia. (Tekla 2016b.)



Kuva 2 Tekla BIMsight käyttöliittymä

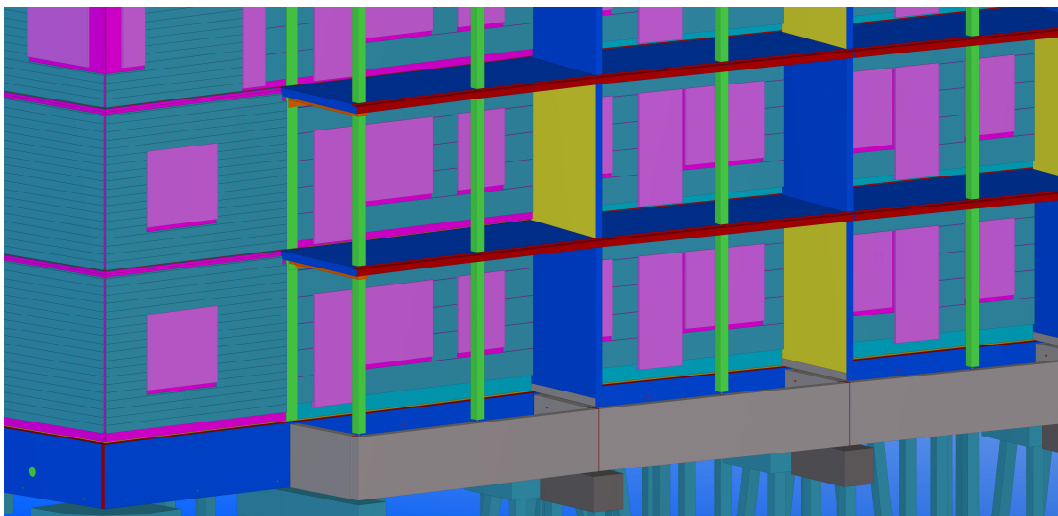
Tekla Structures -ohjelmistolla voidaan laatia mallin osiin kytketyt aikataulut rakentamisen hallintaa varten. Tekla Structures- malliin voi tuoda aikataulun projektihallintaohjelmasta, kuten MS Projectista. Ohjelmiston yhdistäminen toisiin järjestelmiin, muun muassa tuotannon- ja ohjausjärjestelmiin, helpottaa tuotantotiedon siirtämistä valmistajille. Automaattinen tiedonsiirto säästää työaikaa ja auttaa välttämään tarpeettomia virheitä. Mallia voidaan myös hyödyntää määrälaskentaan ja siitä voi ajaa luettelot, joilla tilataan materiaalit työmaalle. (Tekla 2016b.)

2.3 Tekla Structures Full-ohjelmistokokoonpano

Tekla Structures tarjoaa erilaisia lisenssikokoonpanoja, jotka on tarkoitettu eri tyyppisiin projekteihin ja niiden eri vaiheisiin. Tässä kappaleessa esitellään Tekla Structures Full-kokoonpano. Se sisältää kaikki toiminnot, jotka mahdollistavat erilaiset rakennesuunnittelun ja rakentamisen hallinnan toteutukset. Tärkeimmät suunnitelmat, jotka voidaan suunnitella Tekla Structures Full – lisenssin avulla, ovat liitos-, konepaja- ja elementtisuunnittelu.

Seuraavaksi esitellään Tekla Structures Full:in toiminnot ja työkalut paitsi yhteistyötoiminnallisuus, josta on kerrottu edellisessä kappaleessa mainitessa Tekla Structuresin hyödyistä. (Tekla 2016d.)

Mallinnustoiminnoilla voidaan luoda moduuliverkkoja, teräs-, betoni- ja puuosia, hitsausliitoksia, teräskokoonpanoja, betonielementtejä, puuelementtejä, betoniraudotteita sekä erilaisia teräs- ja betoniliitoksia. Mallin osille annetaan erilaisia ominaisuustietoja, kuten materiaali ja numerointisarja. Mallinnetut osat pystytään numeroimaan yksilöllisesti ja malliin voidaan myös lisätä kuormia. Esimerkkinä on kuvassa 3 PuuMERA Honkasuon tietomalli, jossa puuelementit on esitetty vaaleansinisellä ja betonielementit harmaalla värillä. (Tekla 2016d.)



Kuva 3 PuuMERA Honkasuo – tietomalli

Lisäksi kaikkien mallinnustoimintojen jälkeen pystytään tarkastella mallia Organizer-työkalun avulla, joka esitellään kuvassa 4. Organizer-työkalu on mallitiedon jaotteluun ja raportointiin tehty työkalu, jolla voidaan jäsentää ja luokitella mallia ja näyttää mallin puurakenne. Mallin voi luokitella käyttäen esim. numerointisarjoja, statustietoja tai kerros- ja lohkotietoja. Organizerin avulla pystytään helposti löytämään osat, joille on annettu väärää tietoa. Työkalu koostuu kahdesta osiosta Categories (luokat) ja Object browser (kohdeselain). Categories on tarkoitettu mallin luokitteluun ja jaotteluun ryhmiin ja Object browser listaa haluttua tietoa raporttipohjaan. (Juusela 8.3.2016.)

Eräs työkalu, joka on käytössä Tekla Structures Full-lisenssissä, on Task Manager-työkalu ja se on tarkoitettu rakentamisen hallintaan. Task Manager -työkalulla voidaan aikatauluttaa malli ja sen osia voidaan kytkeä aikataulun tehtäviin. Lisäksi tehtävään kuuluville osille voi määrittää asennusjärjestyksen. (Juusela 8.3.2016.)

Object Browser

Name	Content type	Material type	Material	Position number	Profile	Top level / mm	Height / mm	Length / mm	Width / mm	Volume
JÄNNEBE	CAST_UNIT	CONCRETE	Concret	JK-3	RCDL3	7 380	480	5 710	780	
JÄNNEBE	CAST_UNIT	CONCRETE	Concret	JK-4	RCDL3	7 380	480	6 190	780	
JÄNNEBE	CAST_UNIT	CONCRETE	Concret	JK-8	RCL38	7 380	480	6 000	580	
JÄNNEBE	CAST_UNIT	CONCRETE	Concret	JK-5	RCL38	7 380	480	6 190	580	
JÄNNEBE	CAST_UNIT	CONCRETE	Concret	JK-7	RCL38	7 380	580	6 190	580	
JÄNNEBE	CAST_UNIT	CONCRETE	Concret	JK-6	RCL38	7 380	580	5 710	580	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-214	P32	7 409	320	7 821	1 200	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-215	P32	7 409	320	7 821	1 200	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-216	P32	7 409	320	7 821	1 200	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-213	P32	7 409	320	7 821	400	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-217	P32	7 409	320	7 821	1 200	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-214	P32	7 409	320	7 821	1 200	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-214	P32	7 409	320	7 821	1 200	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-218	P32	7 409	320	7 821	1 200	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-212	P32	7 361	320	3 960	800	
ONTELOI	CAST_UNIT	CONCRETE	Concret	O-214	P32	7 409	320	7 821	1 200	
PILARI	CAST_UNIT	CONCRETE	C35/45	P-4	380X38	6 900	380	2 820	380	
PILARI	CAST_UNIT	CONCRETE	C35/45	P-4	380X38	6 900	380	2 820	380	
PILARI	CAST_UNIT	CONCRETE	C35/45	P6-1	380X38	6 900	380	2 820	380	
PILARI	CAST_UNIT	CONCRETE	C35/45	P-4	380X38	6 900	380	2 820	380	
PILARI	CAST_UNIT	CONCRETE	C40/50	P-5	380X38	6 900	380	2 820	380	
PILARI	CAST_UNIT	CONCRETE	Concret	P2-3	580X58	10 500	1 080	10 950	580	
PILARI	CAST_UNIT	CONCRETE	Concret	P2-2	580X58	10 800	1 080	11 250	580	
SA16	ASSEMBLY		S235JRG	EB_SA16-1	D16	4 263	16	10	16	
SA16	ASSEMBLY		S235JRG	EB_SA16-1	D16	4 263	16	10	16	

Number of objects in the table: 27 Result of: Total Of these rows: All 157 297

Kuva 4 Organizer- Tekla Structuresin työkalu

Tärkeimpiä Tekla Structuresin ominaisuuksia on piirustusten ja raporttien luominen. Piirustusten luominen tuottaa erilaisia piirustuksia, joita toimitetaan muille rakennusprojektin osapuolille. Tärkeimpiin piirustustyyppeihin kuuluvat yleispiirustus (taso-, osa- ja asennuspiirustukset), yksittäisen osan ja kokoonpanon piirustus teräsrakenteille, elementtipiirustus betonirakenteille ja paikallavaluraudoitteista piirustus taivutusluetteloineen. Eri piirustuksille voidaan määritellä eri nimiöt toisistaan riippumatta. Raporttien avulla voidaan luoda kokoonpanoluettelot, osaluettelot sekä raudoitusraportteja, jotka sisältävät taivutustaakaaviot raudoituksesta sekä niiden painot ja määrät. Piirustukset ja raportit on mahdollista tulostaa paperille sekä pdf-tiedostoon. (Tekla 2016d.)

3 Tekla Open API

Luku kuvaa Tekla Open API -rajapintaa ja sen käytön tarkoitusta. Alaluvuissa käsitellään rajapinnan käytön hyötyjä ja kerrotaan miten sen avulla on mahdollista tehostaa ja helpottaa Teklan Structuresin käyttöä. Seuraavissa luvuissa käydään läpi Tekla Open API:n kokoonpanoja ja niiden käyttötarkoitusta, kerrotaan Tekla Open API -sovelluksista ja niiden toimintojen riippuvaisuudesta Tekla Structuresiin, kuvataan Plug-in komponentin merkitystä ja sen käytön hyötyjä ohjelmassa ja mainitaan komponentin tyypit ja niiden luomisen vaiheet.

3.1 Yleistä

Tekla on kehittänyt oman avoimen ohjelmointirajapintansa nimeltä Tekla Open API, joka toimii Microsoftin .NET- alustalla. API on lyhyenne sanoista Application Programming Interface, joka tarkoittaa avointa sovellusten ohjelmointirajapintaa. Rajapinnan avulla on mahdollistaa kehittää lisäohjelmia ja sovelluksia Tekla Structures - rakennesuunnitteluohjelmistoon. Tekla Open API toteutetaan Microsoftin .NET- teknologi- alla ja sen avulla kolmannen osapuolen sovellukset pystytään integroimaan samaan ympäristöön. Tekla Open API:n päälle kehitettyjä sovelluksia kutsutaan sovelluslaajennuk- siksi. Laajennukset voivat olla tyyppiä makro, itsenäinen sovellus tai ns. plugin. Laajen- nuksia voi tehdä sekä mallinnus- että piirustustilaan. (Tekla Structures 2016.)

Makro on C#- kooditiedosto, joka käännetään ajettaessa ja liitetään Tekla Structures- pro- sessiin. Makroja voidaan luoda paitsi kirjoittamalla, myös tallentamalla sarjan käyttöliitty- män komentoja. Itsenäinen sovellus toimii nimensä mukaan itsenäisesti, ja kommunikoi Tekla Structures- prosessin kanssa rajapinnan kautta. Plugin on ohjelmistokokoonpano, joka liitetään osaksi Tekla Structures- prosessia Teklan käynnistyksen yhteydessä. Plugin voi olla joko komponentti- tyyppinen, jolloin sillä voi tehdä esim. muokattavan liitoksen tai muun osakokonaisuuden. Plugin voi olla toiminnaltaan myös itsenäisen sovelluksen tyyli- nen, jolloin tuloksena ei synny malliin komponenttia, vaan ainoastaan määrätyt komennot suoritetaan. (Juusela 8.3.2016.)

Tekla Open API tehostaa ja helpottaa Tekla Structuresin käyttöä. Usein toistuvien mekaa- nisten työvaiheiden automatisointi lisää tuottavuutta. Piirustuspuolen esimerkkeinä ovat AutoCAD:in detaljien liittäminen piirustuksiin, mittojen luominen, osamerkintöjen teko tai kokonaisen piirustuksen luominen määrättyjen sääntöjen mukaan. Mallinnuspuolella teho- ja löytyy erilaisten liitosten ja detaljien luomisesta. Rajapinta mahdollistaa myös Tekla Structures -mallin integroinnin muihin ohjelmistoihin, kuten lujuuslaskenta- ja tuotannon-

ohjausohjelmistoihin tai vaikkapa Microsoft Office- toimisto-ohjelmiin. (Tekla Structures 2016.)

Microsoftin .NET- alusta tarjoaa monipuoliset toiminnot mm. käyttöliittymien kehittämiseen. Windows -sovellus on nopea ja miellyttävä luoda, ja uusien asioiden kokeileminen on nopeaa ja tehokasta. Aloittelevakin kehittäjä pystyy kohtuullisella perehtymisellä melko nopeasti jo tekemään hyödyllisiä sovelluksia. (Juusela 8.3.2016.)

3.2 Tekla Open API- rajapinnan osat

Tekla Open API- rajapinnasta löytyy lähes jokaista mallioliota kohti luokka, jonka toiminnallisuudella mallia voidaan muokata. Kaikkien olioluokkien suunnittelu Tekla Open API:ssa noudattaa tiettyjä periaatteita. Objektin alustus ja luonti käyttöliittymässä ovat samankaltaisia, esimerkiksi palkin alustaminen, joka tapahtuu antamalla kaksi pistettä konstruktorimetodissa. Toinen periaate on, että kenttien nimeäminen objektiluokassa noudattaa dialogin kenttien nimeämistä niin paljon kuin mahdollista. Seuraavassa kerrotaan lyhyesti Tekla Open API- rajapinnan kokoonpanojen käyttötarkoituksesta.

(Tekla Structures 2016.)

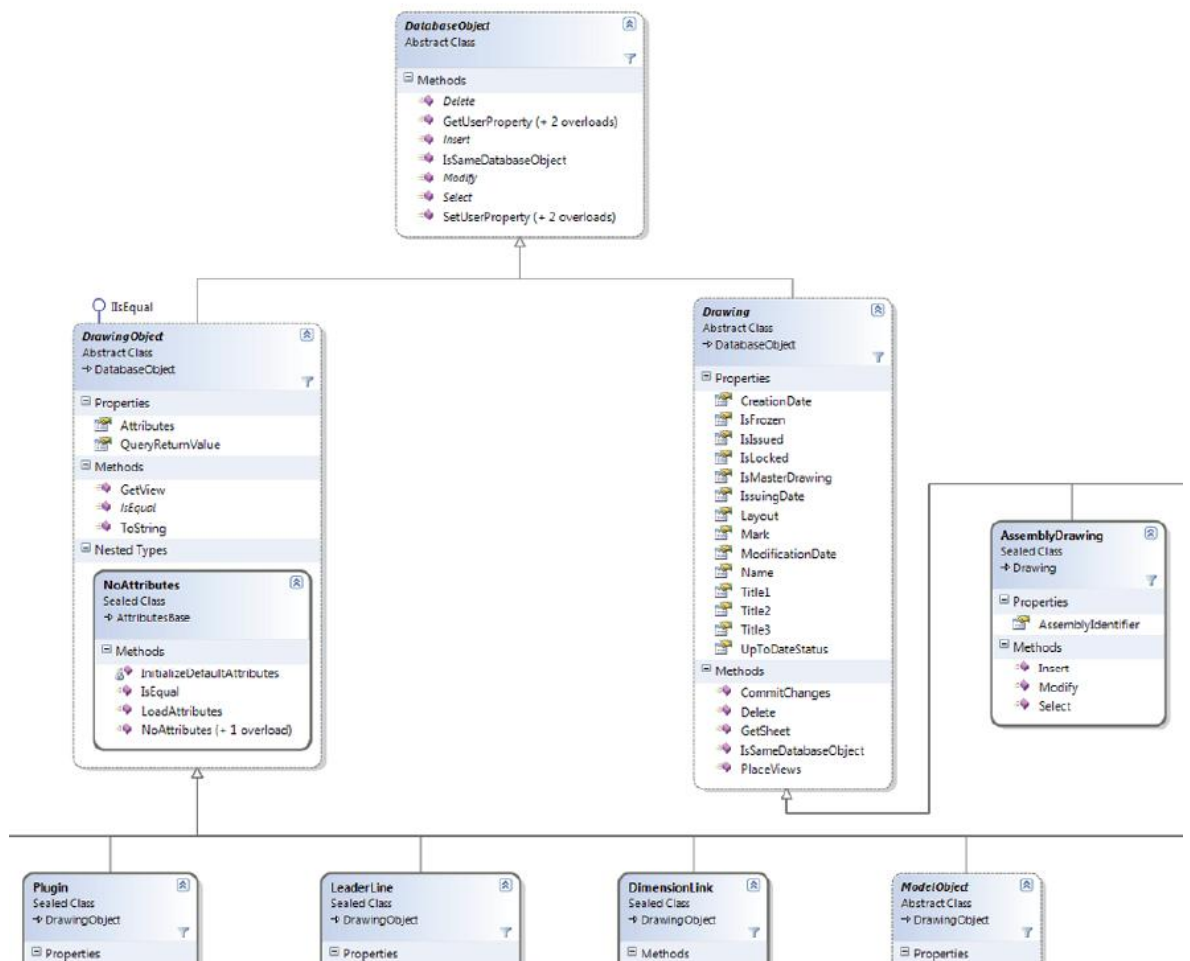
Tekla.Structures.Model.dll -kokoonpano sisältää luokkia ja metodeja mallitietokannan olioiden muokkaamiseen ja yhdistämiseen.

- Malliolio -luokat, kuten Beam, Load, BoltArray, Component...
- Metodit malliolioiden muokkaamiseen, kuten kopiointi, siirto, katkaisu ja liittäminen
- Metodit malliolioiden lukemiseen tietorakenteisiin
- Metodit olioiden ja mallinäkökymien valintaan käyttöliittymästä
- Luokat analyysituloksien tietokantaan liittymiseksi
- Törmäystarkasteluun liittyvät luokat

Tekla.Structures.Drawings.dll-kokoonpano sisältää luokkia ja metodeja piirustustietokannan olioiden muokkaamiseen. Kokoonpanoa suositellaan käytettäväksi piirustusten muokkaamiseen, ei piirustusten luomiseen. Seuraavassa luettelossa esitellään metodien ja luokkien tyyppiä: (Tekla Structures 2016.)

- Piirustusolio -luokat, kuten Drawing, View, Part, Text ja Line
- Metodit malliolioiden lukemiseen tietorakenteisiin
- Metodit olioiden ja mallinäkökymien valintaan käyttöliittymästä

Kuvassa 5 on esitetty osa Tekla.Structures.Drawings- kokoonpanon luokkakaaviosta. (DocFoc 2016.)



Kuva 5 Osa Tekla.Structures.Drawings -kokoonpanon luokkakaaviosta

Tekla.Structures.Plugins.dll-kokoonpano sisältää luokat ja metodit, joita tarvitaan .NET-liitännäisten (plug-in) luomiseen Tekla Structuresiin. Plug-in on periaatteeltaan samanlainen kuin edellisellä Tekla Developer Kitillä luotu ohjelmoitu komponentti. Plug-inin määrittely koostuu käyttöliittymästä, ohjausarvojen hallinnasta, tietorakenteesta sekä varsinaisesta ajosta. Tekla.Structures.Catalogs.dll on kokoonpano, jolla saa yhteyden luetteloihin, kuten profiilit, pultit, materiaalit, raudoitustangot ja tulostimet. Esimerkiksi materiaaliluettelosta pystytään valitsemaan materiaali käyttöliittymän tietokenttään. (Tekla Structures 2016.)

Tekla.Structures.Dialog.dll -kokoonpanon avulla pystytään luomaan käyttöliittymädialogi ja datayhteys Tekla Structures -laajennuksiin. Se myös mahdollistaa dialogin lokalisoinnin ja oletusarvojen tallennuksen. Se tukee Tekla Structuresin tietotyyppisiä ja tarjoaa toimintoja tietotyyppien konversioihin. Tekla.Structures.Datatype.dll -kokoonpano tarjoaa tietotyyppisiä, joiden avulla pystytään kätevästi siirtämään tietoja Tekla Structures -kokoonpanojen välillä. Tietotyypit ovat Distance, Distance List, Boolean, Double, Integer ja String. Tyypit

Distance ja Distance list mahdollistavat muunnokset tuumista milleihin ja päinvastoin. Tekla.Structures.Analysis.dll- kokoonpanon luokat mahdollistavat pääsyn lujuuslaskentatuloksiin. (Tekla Structures 2016.)

3.3 Sovellukset ja plugins

Sovellus voi olla esimerkiksi:

- .NET-sovellus, joka voi olla Windows Forms- tai konsolityyppinen
- COM-sovellus, joka toimii COM- teknologialla
- VBA-makro, joka toimii COM-teknologialla, kuten Microsoft Office -sovelluksissa

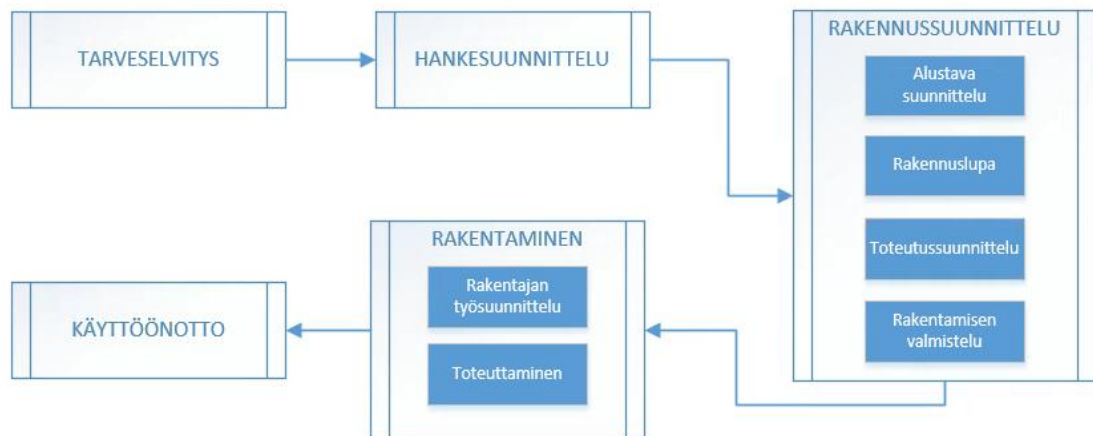
Sovellus käynnistetään erillisenä prosessina. Sovelluksen voi käynnistää Tekla Structuresin käyttöliittymästä tekemällä käynnistystä varten makron, joka kutsuu sitä. Plugin on komponentti-tyyppinen työkalu, joita käytetään mallin tai piirustuksen osien luomisen automatisoimiseen. Pluginit ovat ns. älykkäitä, mikä tarkoittaa sitä, että ne ovat muokattavia sekä riippuvaisia niihin liittyvistä osista. Plugin ladataan osaksi Tekla Structures prosessia, kun Tekla Structures käynnistetään. Malli-plugin käynnistetään Component Catalogilta ja piirustus- plugin työkaluriviltä (engl. toolbar). (Tekla Structures 2016.)

4 Sähkösuunnitelmat osaksi elementtisuunnitelmia

Luvussa selvitetään rakennus- ja elementtisuunnittelun teoriaa. Elementtisuunnittelussa ja toteutuksessa on omat ongelmansa. Huolimaton suunnittelunhallinta ja suunnittelijoiden yhteistyön puute voivat vaikuttaa suunnittelun laatuun ja sen aikatauluun, tässä kappaleessa esitellään sähkösuunnittelun merkitys elementtisuunnittelussa ja eri tavat sähköistyksen lisäämiseen elementtipiirustuksiin.

4.1 Rakennussuunnittelu

Rakennushanke koostuu monista eri vaiheista kuten tarvesuunnittelu, hankesuunnittelu, rakennussuunnittelu, rakentaminen ja käyttöönotto, vaiheet on esitetty kuvassa 6. Hankkeessa on eri osapuolia, kuten rakennuttaja, rakentaja, suunnittelija, käyttäjä ja viranomaiset. (Rakennustieto Oy 1989, 2). Luonnossuunnitelma saadaan hankesuunnittelun pohjalta. Luonnossuunnitelman avulla esitellään tilaajalle ja käyttäjälle yleinen kuvaus hankkeesta ja samalla arvioidaan kustannuksia. Seuraavaksi tehdään rakennussuunnitelma luonnossuunnitelman pohjalta. Sen tekijänä ovat pää-, talo-, ja rakennesuunnittelija (Rakennustieto Oy 1989, 12). Rakennesuunnittelija lähtee luonnossuunnitelmien avulla selvittämään rakennuksen stabiiliteettia, eli miten rakennuksen kohdistuvat kuormat vietään turvallisesti maaperään (Ojanen 11.3.2016).



Kuva 6 Rakennushankkeen vaiheet

4.2 Elementtisuunnittelu

Rakennesuunnittelu sisältää monta eri vaihetta ja yksi niistä on elementtisuunnittelu. Elementtisuunnitteluun vaikuttavat erityisesti päärakennesuunnittelija ja elementtisuunnittelija. Päärakennesuunnittelijan avulla saadaan rakennuksen runko- ja jäykistysjärjestelmän ominaisuudet (lähinnä mitat ja lujuudet) ja niiden vaikutus elementtien mitoittamiseen. Pää-

rakennesuunnittelija antaa kuormituksen lähtötiedot, rungon käyttöiän ja mahdollisesti määrittelee liitos- ja kiinnitysdetaljit. Yleisesti päärakennesuunnittelijan tehtävänä on myös suunnitella ja mitoittaa rakenteet ja liitokset. Ennen elementtien valmistusta päärakennesuunnittelijan on tarkastutettava viranomaisilla tehty elementtisuunnitelmat. Hänen tehtävänsä on myös jakaa tarvittavat tiedot, kuten lähtötiedot, työpohjat sekä reikä- ja varustetiedot muille suunnittelijoille, esimerkiksi reikä- ja sähkösuunnittelijaa varten.

(Elementtisuunnittelu 2016a.)

Elementtisuunnittelija saa piirustukset ja detaljit lujuuslaskelmineen. Elementtisuunnittelijan tekemät rakenne-, työ- ja mittapiirustukset sekä detaljit lujuuslaskelmineen tulee perustua arkkitehdin ja erikoissuunnittelijoiden suunnitelmiin ja detaljeihin. Elementtisuunnittelijan tulee hoitaa omien lujuuslaskelmiensa toimittaminen rakennesuunnittelijalle ennen elementtien valmistamista. Lujuuslaskelmat tarvitaan myös rakennusvalvontaviranomaisia varten. Elementtisuunnittelija on myös velvollinen tekemään elementtiluettelot ja –kaaviot sekä esittämään liittyvien rakenteiden tartuntojen detaljit paikallavalupiirustuksiin. (Elementtisuunnittelu 2016a.)

Elementtirakentamisen suunnittelu vie enemmän aikaa kuin paikallavalurakentaminen elementtien liitosten, varauksien ja detaljien suunnittelun takia. Työt työmaalla kuitenkin vähenevät elementtirakenteiden ansiosta, ja siten runko saadaan pystytettyä nopeamassa aikataulussa. Sen takia on suunniteltava tarkkaan oikeat ratkaisut, joiden avulla aikataulu pystytään toteuttamaan. (Elementtisuunnittelu 2016a.)

Yleisaikataulu koostuu monista tehtävistä, joiden pohjalta luodaan tehtäväluettelo. Yksi tehtävistä on elementtisuunnittelu, jolle määritellään tarkat aikataulut tuotantopiirustusten ja luetteloiden laatimiseksi tehtaalle. Kaikki tarvittavat tiedot sovitellaan elementtisuunnittelun lähtötietovaiheessa rakennus- ja asennusaikatauluihin, asennusjärjestykseen sekä suunnitteluprosessiin. Lisäksi suunnittelun parempaan hallitsemiseen luodaan piirustusai-
kataulu, eli suunnitelma-aikataulu, joka sopeutetaan yleisaikatauluun. Sen avulla määritellään suunnitelmien vaatimukset, sisältö ja ajoitus, jolloin suunnitelmien tulee olla valmiita. (Elementtisuunnittelu 2016a.)

Elementtisuunnittelun kannalta sovitaan toimitettavien lähtötietojen päivämäärät sekä varusten ja reikien aikataulu muiden erillissuunnittelijoiden kanssa ennen piirustusai-
kataluun luomista. Tässä vaiheessa on todella tärkeää ottaa erikseen huomioon aika, joka tarvitaan elementtien sähköistämisuunnittelun sekä muiden varauksien sijoittelun tekemiseen. Tämä vaihe tehdään tilaajaa ja elementtivalmistajaa varten, jotta aikataulu voidaan suunnitella huolella. Tilaajalle ja elementtivalmistajalle on ilmoitettava viivästyksistä lähtö-

tietojen toimittamisessa, jotta nämä viivästyksistä pystytään ottamaan huomioon aikataulun luomisessa. (Elementtisuunnittelu 2016a.)

Isoissa rakennuskohteissa on suuri määrä elementtejä, joista aika usein monet ovat samanlaisia. Samanlaisille elementeille annetaan sama tunnus ja niille tehdään yksi piirustus, jossa ilmoitetaan elementtien tuotantomäärä. Elementtien samanlaisuuden lopullinen varmistus tehdään tarkistamalla muiden suunnittelijoiden mahdolliset lisäykset elementteihin esimerkiksi reiät yms. Vaikka suunnitellut elementit ovat samanlaisia, eri varausten lisääminen tekee niistä erilaisia, joten niiden tunnuksia eivät voi enää olla samoja. (Elementtisuunnittelu 2016a.)

Toinen tapa piirustusten laatimiseen on tehdä jokaisesta elementistä oma piirustus. Tämä toiminta vie enemmän aikaa ja se aiheuttaa ongelmia tehtaalla, koska ei tiedetä mitkä elementit ovat samanlaisia. Jos tiedettäisiin, voitaisiin käyttää samaa muottikalustoa samanlaisten elementtien valmistukseen. Toisaalta suunnitelmien tekeminen elementtikohdasta mahdollistaa muiden erillissuunnitelmien teon helpommalla tavalla. Jos varausten sijainti samanlaisissa elementeissä on eri kohdassa, tarvitaan molemmista elementistä omat piirustukset. Elementtisuunnittelun lopullinen versio saadaan pääsuunnittelijan viimeisen revision pohjalta. Elementtipiirustukset perustuvat suunnittelijan laatimaan mallielementtiin ja työselitykseen. (Elementtisuunnittelu 2016a.)

Elementtipiirustus koostuu monista osista, joiden avulla muodostetaan selkeä ja helposti luettava kokonaisuus. Elementtipiirustuksen pitää sisältää vaaditut tiedot, kuten reunatekstit, tarvittavat leikkaukset, detaljit sekä näkymät eri suunnista. Reunateksteissa sijaitsevat tiedot muun muassa elementin painosta, käyttöiästä, rasioluokasta, betonipeitteestä ja paloluokasta, jotka vaikuttavat olennaisesti siihen, miten elementti valmistetaan. (Ojanen 11.3.2016.)

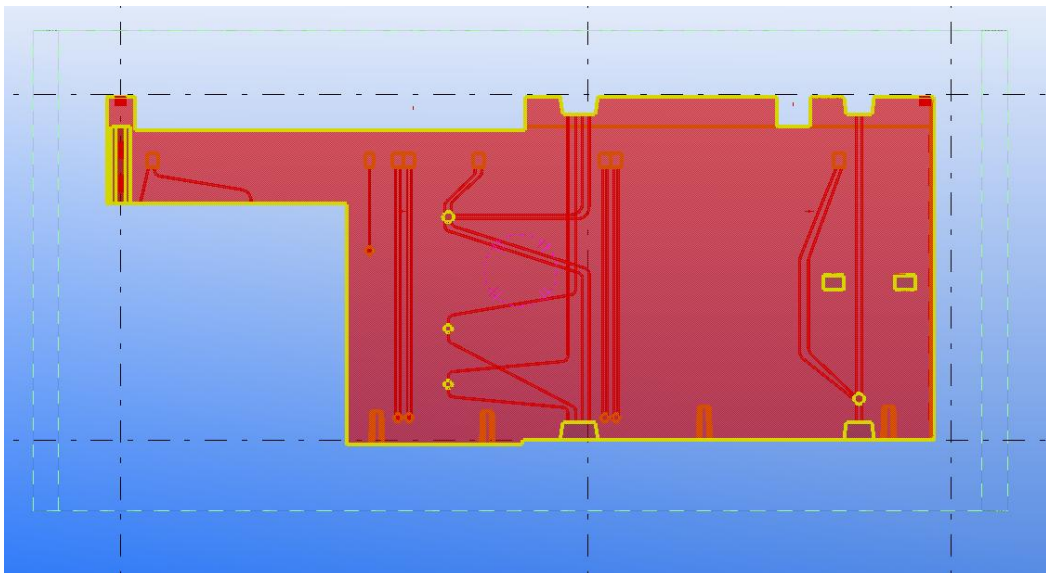
Elementtipiirustuksen laatiminen koostuu monista asioista ja se aloitetaan elementin geometrian määrittelystä. Elementin korkoasema, sijainti, liittymät, syvennykset ja aukot muodostavat elementin muodon. Elementti mitoitetaan piirustuksessa elementin geometrian mukaisesti. Mittojen määrittelyssä käytetään eri viivatyypppejä ja paksuuksia kuvien selkeyttämiseksi. Elementin katsomissuunta on määrittely rakennuksen julkisivuun nähden sisältä ulospäin. Muiden elementtien katsomissuunnan määrittelee tunnuksen lukusuunta. Elementtipiirustuksen ”naamakuvaan” päin katsomissuunta on aina muottipinta ja leikkauskuvien tekeminen tapahtuu aina oikealta vasemmalle tai alhaalta ylöspäin. Yleinen piirustuksien paperikoko on A3 tai suurempi paperiarkki. (Ojanen 11.3.2016.)

4.3 Sähköistysten lisääminen elementteihin

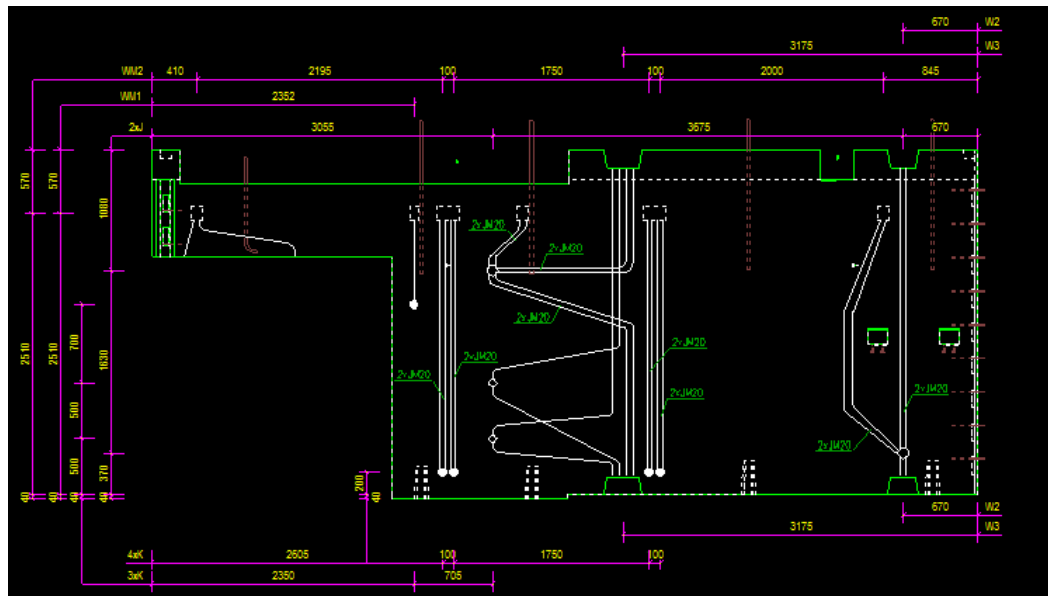
Elementtisuunnittelussa aikataulujen tiukkuus ja dokumenttien iso määrä vaikuttavat lopputulokseen. Suunnittelun nopeuden ja tehokkuuden parantamiseksi käytetään erityisiä ohjelmistoja. Tunnetut ohjelmistot ovat Tekla Structures ja CAD-pohjaiset ohjelmistot, kuten AutoCAD ja CADS Planner House. (Ojanen 11.3.2016.)

Perinteinen tapa sähkösuunnittelussa on sähköistysten lisääminen 2D-piirustuksiin. Prosessi tapahtuu siten, että ensin tehdään elementistä kuva 2D-ohjelmalla, esimerkiksi AutoCAD:lla. Jos elementti mallinnetaan esimerkiksi Tekla Structures ohjelmalla, elementtikuva viedään DWG-tiedostoksi. Seuraavaksi DWG-piirustus lähetetään sähkösuunnittelijalle, joka lisää elementin päälle sähkötarvikkeet. Valmis sähköpiirustus tuodaan Tekla Structuresissa elementtipiirustukseen ja lisätään omalle A3-arkille tai sovittuun näkymään DWG-referenssinä. (Elementtisuunnittelu 2016b.)

Toinen tapa on mallintaa sähköistykset suoraan malliin, mikä mahdollistaa myös tarvikelistan saamisen suoraan mallista. Sähköistys voidaan mallintaa lisäämällä osia manuaalisesti terästyökalujen avulla tai käyttämällä valmiita komponentteja, mikä nopeuttaa työntekoa huomattavasti. Mallinnettuja sähkötarvikkeita mallissa kuvataan kuvassa 7 ja elementtipiirustuksessa kuvassa 8. (Elementtisuunnittelu 2016b.)



Kuva 7 Mallinnettuja sähkötarvikkeita mallissa



Kuva 8 Sähköistys elementtipiirustuksessa

5 Sovellus sähkösuunnitelman liittämiseksi elementtipiirustukseen

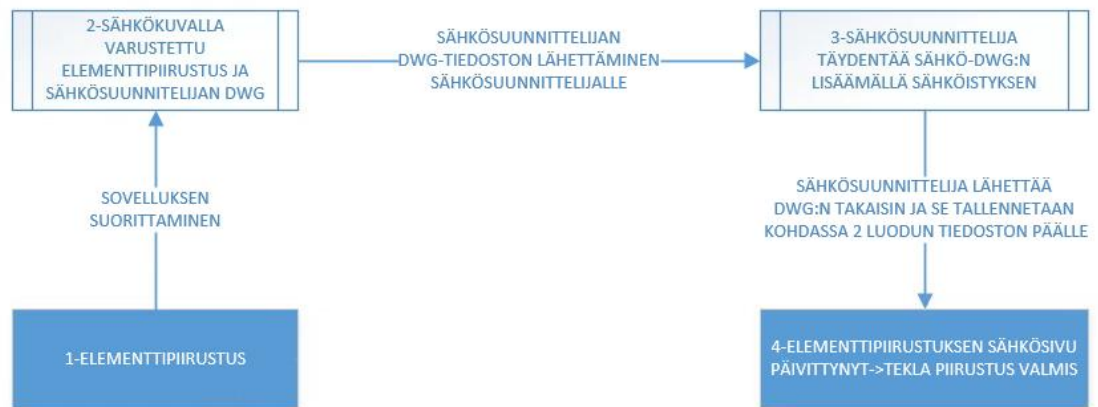
Luvussa kuvataan kehitysprosessi, jolla tuotetaan sovellus sähköistyksen automaattiseksi lisäämiseksi elementtipiirustuksiin. Erikseen esitellään sovelluksen osuus prosessissa. Seuraavaksi kuvataan kaikki toiminnot, jotka on ohjelmoitu sovellukseen. Sovelluksen avulla pystytään luomaan elementistä DWG-piirustus pohja sähkösuunnittelijaa varten ja lisäämään elementtisuunnitelmaan DWG-referenssi tähän piirustus pohjaan. Niiden toimintojen avulla automatisoidaan nykyinen tapa sähköistyksen lisäämiseksi elementtipiirustuksiin.

5.1 Sovelluksen tarve

Sovelluksen luomista ehdotti Sweco Rakennustekniikka Oy:n työntekijä Matti Juusela. Sovellus on tarkoitettu yrityksen Teklan Structures käyttäjille, jotka tarvitsevat uusia ja käteviä työkaluja. Niiden avulla he voivat suorittaa omaa työtään nopeammin ja laadukkaammin. Sweco Rakennustekniikka Oy on myös kehittänyt muita omia sovelluksia ja makroja, joita käytetään päivittäin ohjelmiston yhteydessä. Yleisesti valmiit tuotteet on kehitetty Sweco yrityksen käyttöä varten kuten tässäkin opinnäytetyössä. Yritys voi myös myydä kehittämiään tuotteita toisille yrityksille. Sähköistyksen lisääminen elementtipiirustuksiin kestää tietyn aikaa ja se koostuu monesta toiminnoista, joten kehitystarpeena on automatisoida prosessi. Sovellus käsittelee valittuja elementtipiirustuksia elementtien piirustusluettelosta. Projektin luomisen vaiheet esitellään tarkemmin seuraavassa luvussa.

5.2 Prosessi

Projektin toteutus koostuu monista vaiheista. Jokainen toiminto, jonka Tekla Structures -käyttäjä piti suorittaa manuaalisesti, yritettiin automatisoida. Sovelluksen luonti tapahtui Tekla Structures -käyttäjien mielipiteiden ja arviointien sekä sähkösuunnittelijoiden vaatimuksien perustella. Seuraavaksi kuvataan sähköistyksen lisäämisprosessin vaiheet, jotka on esitetty kaaviona kuvassa 9.



Kuva 9 Automatisoitu sähköistyksen lisääminen prosessi vaiheet

Prosessin vaiheet:

- Tehdään elementtipiirustuksista DWG-pohjat sähkösuunnittelijaa varten
- Lisätään sähkösuunnittelijan DWG- pohjat elementtipiirustuksiin DWG- referensseinä
- Lähetetään DWG- pohjat sähkösuunnittelijalle
- Sähkösuunnittelija lisää sähkösuunnitelmat DWG- pohjiin
- Tallennetaan sähkösuunnittelijan täydentämät DWG-pohjat edellisten päälle, jolloin elementtipiirustukset päivittyvät automaattisesti Tekla Structuresissa.
- Elementtisuunnittelija tarkastaa piirustuksen.

Kaikki toiminnot ohjelmoidaan AddElectricDWG-sovelluksessa ja toimintojen tarkempi kuvaus esitellään jäljempänä. Seuraavassa luettelossa esitellään toiminnot:

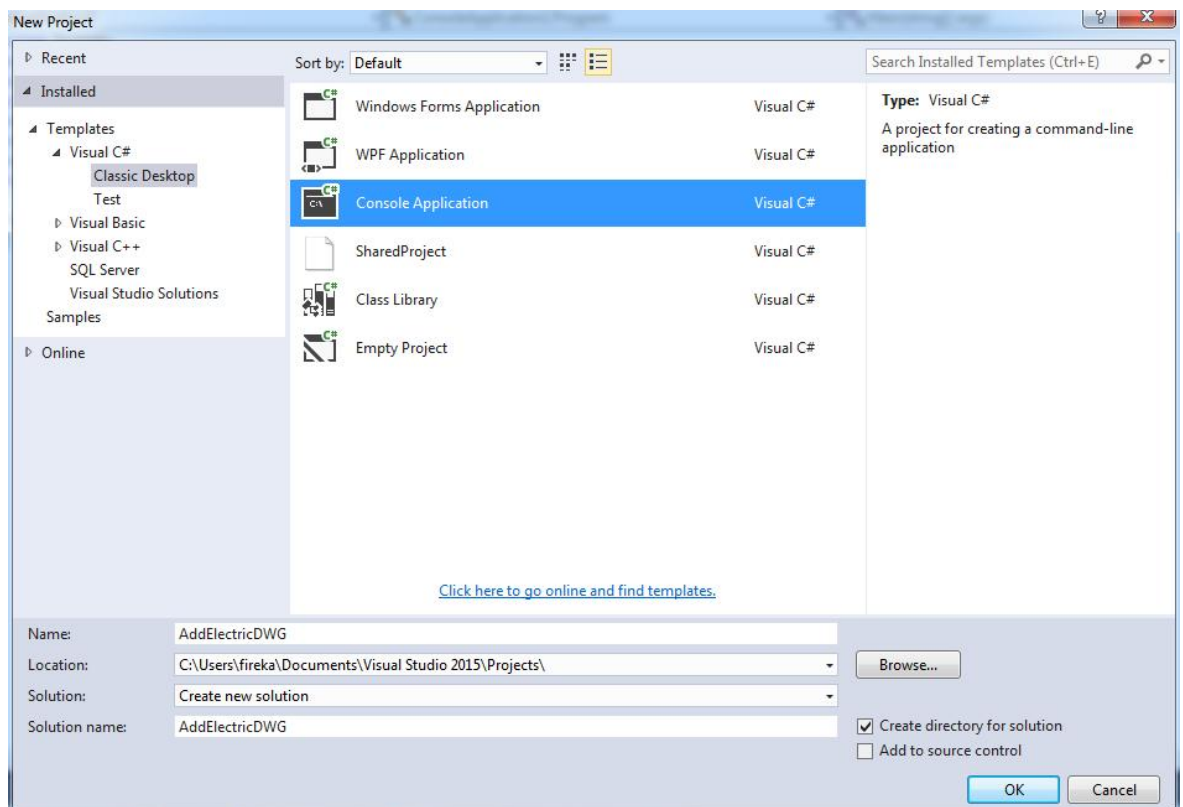
- piirustuksen avaaminen
- ylimääräisten näkymien ja objektien (esim. mitat, teksti) poistaminen piirustuksesta
- tarvittavien näkymien siirtäminen yhdelle paperiarkille ja muiden arkkien poistaminen
- sähkösuunnittelija -layoutin lataaminen piirustukseen
- piirustuksen vienti DWG-tiedostona mallin alle luotuun kansioon
- piirustuksen sulkeminen tallentamatta
- piirustuksen avaaminen uudelleen
- paperiarkin lisääminen sähkösuunnitelmaa varten
- äsken viedyn DWG-piirustuksen lisääminen paperiarkille, joka on tarkoitettu sähköistykseen

5.3 Sovelluksen kehittäminen

Sovelluksen kehittäminen aloitettiin tammikuun 2016 puolivälissä ja kesti kolme kuukautta. Olen toteuttanut projektin yksin ohjaajan valvonnassa. Projektin menetelmä pääosin koostui palaverista, joka pidettiin kaksi kertaa viikossa, jossa keskusteltiin projektin etenemisestä ja erilaisien ongelmien ratkaisusta. Projektin alkuvaiheessa piti tutustua myös erilaisiin dokumentteihin, jotta sain pohjatiedot sovelluksen toteuttamiseen. Dokumentaatio oli hyvä lähtökohta oppimiseen ja sovelluksen kehittämiseen. Hankin myös ratkaisuja ongel-

miin sovelluksen kehittämisessä internetistä pääosin Tekla Open API forumista etsien samankaltaisia ongelmia, joihin törmäsin omassa projektissani. Lisäksi olen hyödyntänyt ammattilaisten mielipiteitä, muun muassa Teklan Structures käyttäjiltä, elementtisuunnittelijoilta ja sähkösuunnittelijoilta, joita olen haastattelut sovelluksen toteuttamisen aikana. Kerättyjen tietojen perustella löysin ongelmiin ratkaisuja. Projektin toteuttamisessa oikea ratkaisu löytyi kokeilemalla yrityksen ja erehdyksen kautta. Sovelluksen toimivuuden tarkastamista varten olen käyttänyt testimallia, joka on toiminut sovelluksen yhteydessä rajapinnan kautta. Sovelluksen toimivuus oli testattu, mutta systemaattinen testauksen kuvaaminen on rajattu työstä pois.

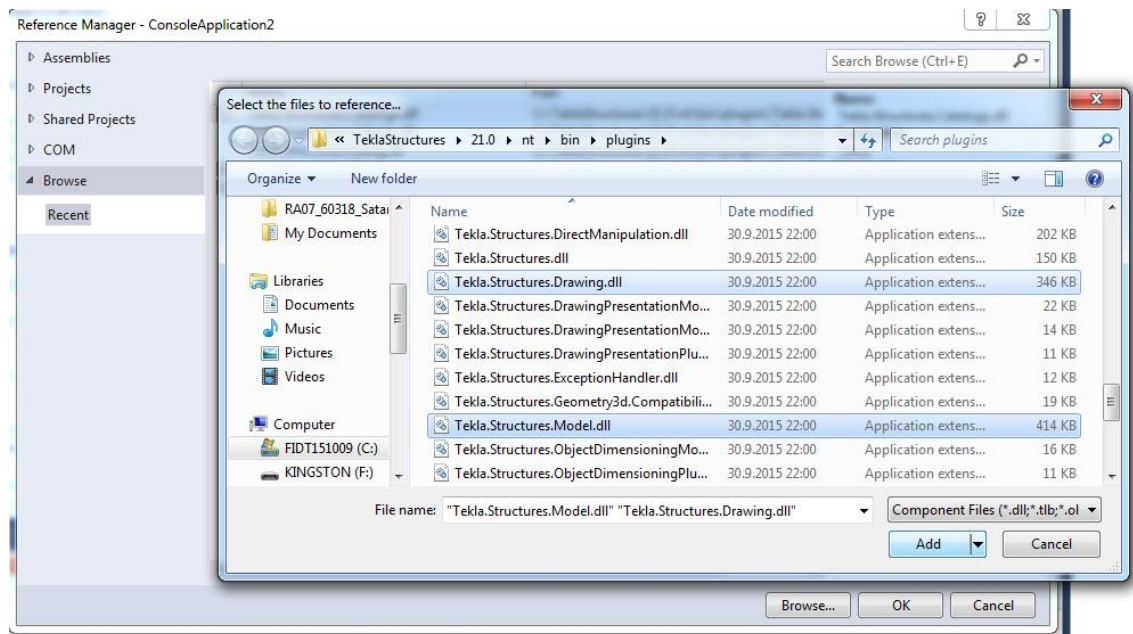
Sovelluksen toteuttamiseen käytetään Microsoft Visual Studio 2015 kehitysympäristöä ja ohjelmointikielenä on C#. Sovelluksen luomisen rajapintana käytetään Tekla Open API-ohjelmointirajapintaa. Sovellusta varten luodaan uusi projekti Visual Studiassa, joka nimitetään AddElectricDWG:ksi. Seuraavaksi valitaan Visual C#:in Console Application-tyyppi, jolle annetaan nimi AddElectricDWG, kuten on kuvattu kuvassa 10.



Kuva 10 Uuden projektin luominen

Projektin luomisen jälkeen sovellukselle tehdään toimenpiteet, jotta sovelluksen luokkarakenne saadaan toimimaan oikealla tavalla. Toinen toiminto on sovelluksen tietojen välitys isäntäohjelmistoon ja sen jälkeen otetaan yhteyttä Tekla Open API:iin lisäämällä tarvitta-

vat referenssit, joiden avulla voidaan kutsua tarpeellisia objekteja kirjastoista. Referenssi-kirjastot, joita tarvitaan sovelluksen luomisen ovat Tekla.Structures.Model.dll, Tekla.Structures.Model.Operations.dll, Tekla.Structures.Drawing.dll, Tekla.Structures.Drawing.UI.dll ja Tekla.Structures.Geometry3d.dll. Niiden lisääminen on kuvattu kuvassa 11.



Kuva 11 Referenssien lisääminen

Ennen piirustusten käsittelyä käytetään Drawing Handler -luokkaa, joka alustaa yhteyden piirustuksiin Tekla Structuresissa. Ilman Drawer Handler -luokkaa ei ole mahdollista suorittaa toimintoja piirustuksiin. Luokan avulla saadaan luettua piirustuslista tai vaikka luotua uusi piirustus. Piirustuksen avaamiseen ja piirustuksessa olevien objektien käsittelyyn tarvitaan tietyt luokat. Luokka, joka tarjoaa pääsyn piirustusluetteloon, on Drawing Selector ja piirustukset luetaan DrawingEnumerator-tietorakenteeseen. Model-luokan ilmentymän luonti luo yhteyden Tekla Structures malliin. Kaikki edellä mainitut luokat esitellään alla olevilla komennoilla:

```
DrawingHandler drawingHandler = new DrawingHandler();
DRUI.DrawingSelector drSelector = drawingHandler.GetDrawingSelector();
DrawingEnumerator drawingEnum = drSelector.GetSelected();
Model teklaModel = new Model(); //yhdistetään avoimeen malliin
```

5.4 Piirustusten avaaminen ja käsittely

Aloitustoimintojen jälkeen avataan ja käsitellään ainoastaan yhtä piirustusta kerrallaan. Vaikka piirustusluettelossa on valittu monta piirustusta sovellus suorittaa toimintoja vain

ensimmäiselle piirustukselle. Näin varmistetaan, että useassa piirustuksessa ei tapahdu hallitsemattomia muutoksia. Kun piirustus on auki, ajetaan sovellus pelkästään sille, vaikka luettelossa olisi valittuna muitakin piirustuksia. Mikäli sovelluksen annettaisiin muokata ja tallentaa piirustuksia automaattisesti, voisi joissain piirustuksissa esiintyä ongelmia, jotka on korjattava jälkikäteen. On parempi, että muutokset ovat käyttäjän tarkastettavissa ennen tallennusta, vaikka olisi erittäin tehokasta ajaa kaikki piirustukset kerralla.

Sovellus käy läpi valittujen piirustuksien aktiiviset näkymät, tarkistaa aktiivisen näkymän nimet ja sen perustella poistaa kaikki tarpeettomat näkymät. Jos piirustuksessa on monta näkymää samalla nimellä, poistetaan kaikki paitsi se, jota on käsitelty ensimmäisenä. Seuraavaksi sovellus käsittelee näkymät, jotka ovat jääneet piirustukseen. Sähkösuunnittelun DWG-piirustusta varten tarvitaan pelkästään elementin päämitat ja näkymän nimi, kaikki muut objektit poistetaan. Käsitelty näkymä siirretään ensimmäiselle paperiarkille, josta luodaan DWG-tiedosto sähkösuunnittelijaa varten. Alla olevassa koodissa esitellään edellä mainitut sovellusten toiminnot:

```
if (drawingEnum.MoveNext()) // ottaa vain ensimmäinen piirustus valituista
//while (drawingEnum.MoveNext()) //voidaan ajaa monta piirustusta

{
    Drawing drawing = drawingEnum.Current;

    if (drawingactive != null)
        //jos piirustus auki ajetaan pelkästään sovellusta aukeavaan piirustukseen
    {
        drawing = drawingactive;
    }
    else
    {
        drawingHandler.SetActiveDrawing(drawing, true);
        // muuten ajetaan ensimmäinen valittu piirustus luettelosta
    }

    DrawingObjectEnumerator views = drawing.GetSheet().GetObjects();

    bool frontFound = false;
    bool topFound = false;
    bool bottomFound = false;
    bool backFound = false;

    while (views.MoveNext())//käydään läpi näkymät
    {
        View view = views.Current as View;

        if (view != null)
        {
            tiidot.Add(view, view.Origin);

            if (view != null && !view.Name.Equals("FRONT") && !view.Name.Equals("TOP") &&
            !view.Name.Equals("BOTTOM") && !view.Name.Equals("BACK"))
            {
                views.Current.Delete(); //poistetaan tarpeettomat näkymät
            }

            if (view.Name.Equals("FRONT"))
```

```

{
    if (frontFound == true)
        views.Current.Delete(); //poistetaan toistuvat näkymät
    else
        frontFound = true;
}
//yllä oleva bool toiminto käytetään muille näkymille

DrawingObjectEnumerator viewObjects = view.GetObjects();

while (viewObjects.MoveNext())
{
    if ( viewObjects.Current is Mark || viewObjects.Current is MarkSet ||
        viewObjects.Current is ReinforcementBase || viewOb-
        jects.Current is SectionMarkBase)
        viewObjects.Current.Delete(); //poistetaan tarpeettomat objektit
}

int N = (int)view.Origin.Y / 297;

if (N > 0)
//if (view.Origin.Y > 297 && view.Origin.Y < 594)
{
    double originX = view.Origin.X;
    double originY = view.Origin.Y;

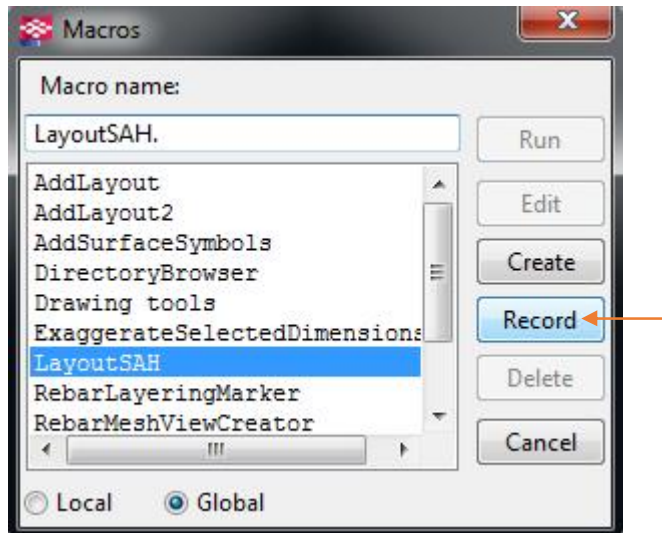
    view.Origin = new Point(originX, originY - (N*297), 0);
    view.Modify();
}
}

```

Lopuksi ajetaan tähän asti tehdyt muutokset piirustukseen komennolla:

```
drawing.CommitChanges();
```

Seuraavaksi luodaan käyttöliittymämakro Tekla Structuresissa, jonka avulla pystytään lataamaan sähkösuunnittelijan layout. Tämä toiminto suoritetaan painamalla Record-painike Macros-paneelissa. Makro luodaan nauhoittamalla kaikki toiminnot, joiden avulla saadaan ladattua uusi layout piirustuksiin. Makron käynnistys on esitelty kuvassa 12.

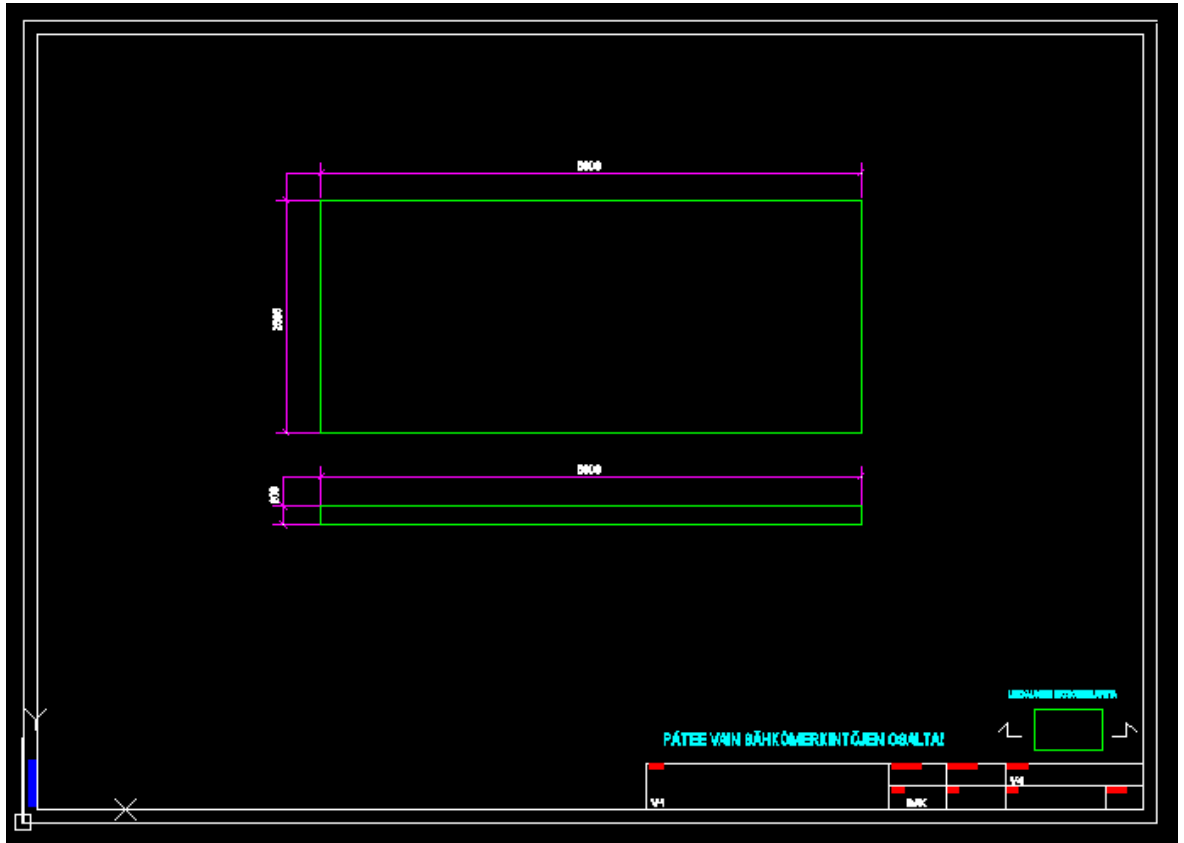


Kuva 12 Makron nauhoittaminen Tekla Structuresissa

LayoutSAH luodaan Tekla Structuresin Layout- editorilla, joka mahdollistaa esim. raamien, luetteloiden ja nimiön lisäämisen ja sijoittelun. Sähkösuunnittelijaa varten layouttiin lisätään tarvittavat tekstit ja leikkauksen katsomissuunta, joiden avulla sähkösuunnittelija pystyy lisätä sähköistykset oikeisiin paikkoihin. Tallennetun makron lisääminen sovelluksessa tapahtuu seuraavalla komennolla:

```
Operati on. RunMacro(@".. \drawi ngs\LayoutSAH. cs");
```

Edellä mainituilla toiminnoilla saadaan sähköpiirustus pohja, joka viedään DWG-tiedostoksi. Valmis piirustus pohja sähkösuunnittelijaa varten on esitelty kuvassa 13.



Kuva 13 Elementtipiirustus sähkösuunnittelijan varten

5.5 Piirustusten vienti DWG-tiedostona

Ennen sähköpiirustus pohjan vientiä luodaan kansio, mihin DWG-tiedosto voidaan tallentaa. Seuraavalla komennolla suoritetaan tämä toiminto:

```
string model path = teklamodel.GetInfo().ModelPath; //mallin polku

if (!Directory.Exists(model path + @"\SAH"))
{
    Directory.CreateDirectory(model path + @"\SAH"); //polun luominen
}
```

ExportDWG on metodi, jolla sovellus vie DWG-tiedoston luotuun SAH-kansioon. Metodissa luodaan Tekla Structures -makro CreateDWG.cs. Makron ”pohja” on luotu nauhoittamalla käyttöliittymämakro. Sen pohjalta on tehty koodi, joka luo makron ”lennosta”. Makron suorittamisen jälkeen poistetaan CreateDWG.cs. Edellä mainitut toiminnot suoritetaan seuraavilla komennoilla:

```
private static void ExportDWG(Drawing drawing)
{
    string makrocontent = "";
    string quote = ' '.ToString();
    drawing.Select(); //piirustuksen valinta
    string nimi = drawing.Mark;
    //makrocontentin sisältö macron pohjalta tässä kohdassa
```

```

Model teklaModel = new Model(); //yhdistetään avoimeen malliin
string macropath =
    @"C:\TeklaStructures\21.0\Environments\Sweco_FI\00_SWECO_COMMON\01_Sweco_Common\Macros\drawings"; //mallin polku

StreamWriter writer = File.CreateText(macropath + @"\CreateDWG.cs");
writer.Write(makrocontent);
writer.Close();
Operation.RunMacro(@"..\drawings\CreateDWG.cs");
File.Delete(macropath + @"\CreateDWG.cs");
}

```

ExportDWG-metodia kutsutaan Main-metodissa. Metodin kutsu Main-metodissa tapahtuu seuraavasti:

```
ExportDWG(drawing);
```

DWG-tiedoston viennin jälkeen suljetaan alkuperäinen elementtipiirustus tallentamatta, koska siihen ei haluta tässä vaiheessa tehdä minkäänlaisia muutoksia. Vasta kun lisätään vastanotettu sähkösuunnittelijan DWG-tiedosto, tehdään tallennettavat muokkaukset piirustuksiin. Komento, joka mahdollistaa kuvan sulkemisen tallentamatta on:

```
drawingHandler.CloseActiveDrawing();
```

5.6 Sähköpiirustuspohjan lisääminen DWG- referenssinä

Sähkösuunnittelija lähettää täydennetyn DWG-tiedoston sähköistyksellä takaisin elementtisuunnittelijalle. Seuraavaksi suoritetaan AddLayout-metodi, jonka avulla avataan uudeleen elementtipiirustus ja lisätään paperiarkki sähkömerkintöjä varten sekä lisätään viittaus sähköpiirustukseen äsken lisätylle paperiarkille. Paperiarkin lisääminen tapahtuu suurentamalla yhdellä paperiarkin korkeudella nykyistä elementtipiirustusta.

Muutoksien jälkeen haetaan sähkösuunnittelijan DWG-tiedosto, joka on nimetty elementtittunnuksen perustella. Jos kansiossa SAH on olemassa haettava sähköpiirustus, elementtipiirustukseen lisätään viittaus siihen ja kohdistetaan sähkösuunnittelijan DWG-tiedosto lisätylle paperiarkille. Lisäksi muutetaan DWG-tiedoston raamit näkymättömäksi, että ne eivät tule näkyviin elementtipiirustuksessa. Kaikki nämä toiminnot on tehty metodissa addLayout():

```

private static void addLayout(Drawing drawing)
{
    DrawingHandler drawingHandler = new DrawingHandler();
    DRUI.DrawingSelector drSelector = drawingHandler.GetDrawingSelector();
    Model teklaModel = new Model();
    drawingHandler.SetActiveDrawing(drawing, true);
}

```

```

// piirustuksen avaaminen uudelleen

double Width = drawingHandler.GetActiveDrawing().Layout.SheetSize.Width;
double Height = drawingHandler.GetActiveDrawing().Layout.SheetSize.Height;

drawing.Layout.SheetSize.Height = Height + 297; //paperiarkin lisäminen
drawing.Modify(); //muokata piirustus
drawing.CommitChanges(); //siirtää muutokset
drawing.Select(); //piirustuksen valinta
string nimi = drawing.Mark; //piirustuksen numero
nimi = nimi.Replace("[", "").Replace("]", "").Replace("/", "-").Replace(".", "-").Replace(" ", "-").Replace(" ", "");
string nimi2 = teklaModel.GetInfo().ModelPath + @"\\SAH\\" + nimi + "_sah.dwg";

try
{
    DwgObject dwgObject = new DwgObject(drawing.GetSheet(), new
    Tekla.Structures.Geometry3d.Point(-10.5, Height + 7.43, 0), nimi2);
    dwgObject.Attributes.Frame.Color = DrawingColors.Invisible;
    dwgObject.Insert();
}
catch (Exception exp)
{
}
}

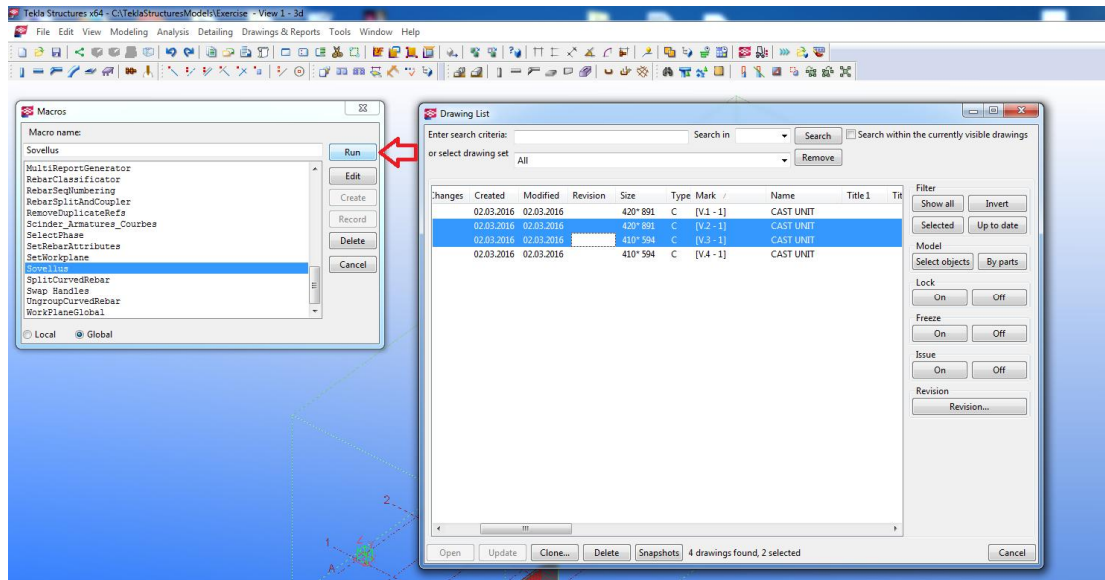
```

Metodin kutsu Main()-metodissa:

```
addLayout(drawing);
```

DWG-tiedoston lisäämisen jälkeen ei suoriteta tallennusta tai piirustuksen sulkemista, koska käyttäjän tulee tarkistaa piirustus mahdollisten vaurioiden varalta. Jos DWG-tiedoston lisääminen onnistui, käyttäjä sulkee piirustuksen tallentaen sen. Käyttäjän on tarkasteltava piirustus, sillä tietokoneohjelmiin ylipäänsä ei kannata luottaa sokeasti. Virheitä voi syntyä koneella paljon ja nopeasti, jos käytännön testeissä alkaa näyttää siltä, että vaurioita ei tule, voidaan harkita automaatioasteen nostamista eli ajaa sähköpohjat ja referenssit kaikkiin valittuihin piirustuksiin kerralla.

Sovellus käsittelee valittuja elementtipiirustuksia elementtien piirustusluettelosta. Sovellus käynnistetään painamalla "Run"-painiketta, joka on esitetty kuvassa 14.



Kuva 14 Sovelluksen käynnistys Tekla Structuresissa

6 Yhteenveto

Rakennusala kehittyy koko ajan ja sitä varten tarvitsee kehittää ohjelmistoja, joiden avulla saadaan aikaan uusia ratkaisuja ja parantaa toimintoja. Varsinkin rakennussuunnittelun ohjelmistojen on kehityttävä, jotta saavutetaan asiakkaiden tarpeiden mukaan parempi lopputulos. Opinnäytetyön tavoitteena oli luoda sovellus Sweco Rakennetekniikka Oy:lle, jonka avulla voidaan automatisoida Tekla Structures – ohjelman yksi perustoiminto, mikä nopeuttaa työskentelyä ja välttää tarpeettomia virheitä. Opinnäytetyössä kehitetylle sovellukselle oli tarvetta, sillä sen avulla tulevien toimeksiantojen elementtisuunnittelusta saadaan huomattavasti tehokkaampaa ja näin yritys pystyy vastaamaan koko ajan kiristuviin aikatauluihin.

Perustoiminto, jota käsiteltiin tässä työssä, on sähköistyksen lisääminen elementtipiirustukseen. Teoriaosuudessa esiteltiin asiat, jotka vaikuttivat sovelluksen toteuttamiseen. Sovelluksen kehityksessä rajattiin aihe sen luomiseen eikä testaukseen ja käyttöönottoon, jotka suoritetaan myöhemmin Swecon toimesta. Kehitystyö onnistui ja sovelluksen toimivuus on tarkastettu koodaamisen aikana. Kehittynyt työmenetelmä on huomattavasti helpottanut ja nopeuttanut sähköistyksen lisäämistä elementtipiirustukseen, mikä parantaa elementtisuunnittelun tehokkuutta ja sitä kautta tuo lisäarvoa yritykselle.

Sovelluksen kehittämisessä käytettiin Microsoft Visual Studio 2015 -kehitysympäristöä ja sovellusten ohjelmointikielenä oli C#. Sovelluksen toteuttamiseksi valittiin Visual C#:in Console Application -pohja, johon luotiin sovelluksen kaikki koodin osat, muun muassa metodit, luokat ja kirjastot. Sovelluksen kehittäminen Tekla Structures -ohjelmistoon oli mahdollista Teklan oman avoimen Tekla Open API -ohjelmointirajapinnan avulla, joka toimii Microsoftin .NET -alustalla. Sovellus toimii itsenäisesti ja se kommunikoi rajapinnan kautta Tekla Structures-ohjelmiston kanssa käynnistyksen yhteydessä.

Aikaisempi työmenetelmä, jossa sähkösuunnitelmien liittäminen elementtipiirustuksiin koostuu monista vaiheista, oli aikaa vievä ja työläs prosessi. Jokainen prosessiin kuuluva toiminto piti suorittaa erikseen. Tarkoituksena oli automatisoida toiminnot, jotka käyttäjän tuli suorittaa manuaalisesti. Toiminnot ohjelmoitiin tiettyjen metodien avulla ja sovelluksen avulla käyttäjän ei tarvitse enää suorittaa yksittäisiä toimintoja itse, vaan sovellus hoitaa ne hänen puolestaan.

Opinnäytetyön tuloksena luotu sovellus mahdollistaa sähköistyksen lisäämisen elementtipiirustukseen. Jatkossa sovellus voidaan kehittää käyttämään automatisoitua prosessia monelle elementtipiirustukselle yhtä aikaa. Sovelluksen jatkekehitys nopeuttaisi prosessia

vielä enemmän kuin nykytilanteessa, koska elementtisuunnittelijan ei tarvitse tarkistaa jokaista piirustusta erikseen, sillä sovellus tekisi muutokset valittuihin elementtipiirustuksiin automaattisesti. Kehitetty sovellus on yksi monista tuotteista, jotka on kehitetty Swecolle. Yritys kehittää myös muita erilaisia työkaluja, joiden avulla he pystyvät tarjoamaan vaativia tietomallinnuskomponentteja omille käyttäjille tai ulkopuolisia tilauksia varten. Sweco Oy panostaa Teklan Structuresin omien työkalujen kehittämiseen, koska ohjelmisto on johtava tuote markkinoilla ja se edistää huomattavasti rakennesuunnittelua.

Projektin aikana olen oppinut todella paljon uusia asioita. Olen suorittanut työharjoittelun Sweco Oy:lle ennen opinnäytetyön kirjoittamista, joten pystyin hyödyntää opittuja ohjelmistotaitoja opinnäytetyössä. Ennen harjoittelua minulla ei ollut oikeastaan aiempaa kokemusta Microsoft Visual Studion ja Tekla Open API:n käytöstä. Tekla Structures kokemusta oli hankittu aiemmista suunnitteluassistentin tehtävistä.

Projektin aikana olen oppinut elementtisuunnittelusta paljon ja pystyn ymmärtää elementtisuunnittelun prosessia selkeämmin kuin ennen. Oli todella hyvä asia, että pystyin tutustua ja oppia ymmärtämään Tekla Open API:n menetelmää työharjoittelun aikana, koska sen toimintojen ymmärtäminen vie aikaa ennen kuin pystytään luoda omia sovelluksia. Haasteena oli ohjelmointitaidon opettelu, joka kesti kauan. Projektin toteuttaminen vaatii osaamisen kerryttämistä tutustumalla ohjelmaan ja harjoittelemalla ohjelman käyttöä, joten tutustuin perus komponenttien ja sovelluksien luomiseen ja toimivuuteen. Vaikka uuteen asiaan tutustuminen oli vaikeaa alussa, olen tyytyväinen, koska rajapinnan ja Visual Studion avulla pystytään kehittämään vaativia työkaluja tai sovelluksia Tekla Structures-ohjelmistoon. Tarkoituksena tulevaisuudessa on laajentaa Tekla Open API:n osaamista ja kehittää uusia sovelluksia Swecolla. Projektin aikana huomasin, että laajensin runsaasti Tekla Structures taitojani. Sovelluksen luominen vaati paljon Teklan Structuresin perustietoa, joten minun piti tutkia sen toimintoja, esimerkiksi elementtipiirustuksen nimiön luontia.

Lähteet

DocFoc 2016. TeklaOpenAPI-17. Luokka kaaviot. Helsinki. Luettavissa:
<http://www.docfoc.com/teklaopenapi-17-classdiagrams-a0>. Luettu: 1.4.2016.

Elementtisuunnittelu 2016a. Suunnitteluprosessi. Mallintava-suunnittelu. Helsinki. Luettavissa: <http://www.elementtisuunnittelu.fi/fi/suunnitteluprosessi/mallintava-suunnittelu>.
Luettu: 11.3.2016.

Elementtisuunnittelu 2016b. Suunnitteluprosessi. Mallintava-suunnittelu. BEC2012 Elementtisuunnitteluohje_v104. Helsinki. Luettavissa:
http://www.elementtisuunnittelu.fi/Download/23982/BEC2012%20Elementtisuunnittelun%20mallinnusohje_v104.pdf. Luettu: 11.3.2016.

Juusela, M. 8.3.2016. Teknologiapäällikkö (BIM). Sweco Rakennetekniikka Oy. Haastattelu. Helsinki.

Ojanen, S. 11.3.2016. Suunnitteluavustaja. Sweco Rakennetekniikka Oy. Haastattelu. Helsinki.

Rakennustieto Oy 1989. Kortistot. Tuotteet. RT_180. RT 10-10388. Talonrakennushankkeen kulku. Luettavissa:
<https://www.rakennustieto.fi/bin/get/id/5guoZSPW8%3A%2447%2410387%2446%24pdf.0.0.5gunJ4yOi%3A%2447%24handlers%2447%24net%2447%24statistics%2495%24download%2495%24pdf%2446%24stato.5gv06pzjY%3AC1-RT%2495%24180/10387.pdf>
Luettu: 11.3.2016.

Tekla 2016a. Tietoa meistä. Lyhyesti. Luettavissa: <http://www.tekla.com/fi/tietoa-meista/lyhyesti>. Luettu: 8.2.2016.

Tekla 2016b. Tuotteet. Tekla-Structures. Luettavissa:
<http://www.tekla.com/fi/tuotteet/tekla-structures>. Luettu: 10.2.2016.

Tekla 2016c. Ratkaisut. Rakennusliikkeet. Luettavissa:
<http://www.tekla.com/fi/ratkaisut/rakennusliikkeet>. Luettu: 8.2.2016.

Tekla 2016d. Tuotteet. Tekla Structures. Ohjelmistokokoonpanot. Tekla-structures-full. Kaikki-toiminnallisuudet. Luettavissa: <http://www.tekla.com/fi/tuotteet/tekla->

structures/ohjelmistokokoonpanot/tekla-structures-full-%E2%80%93-kaikki-toiminnallisuudet. Luettu: 12.2.2016.

Tekla BIMsight 2016. Teklabimsight. Learn-more. What-is-tekla-bimsight. Luettavissa: <https://www.teklabimsight.com/learn-more/what-is-tekla-bimsight>. Luettu: 12.2.2016.

Tekla Structures 2016. Tekla open api. Luettavissa: https://teklastructures.support.tekla.com/201/en/sys_tekla_open_api. Luettu: 15.2.2016.

Tekla Warehouse 2016. Intranet. Tekla Open API StartUp Package. Luettu: 12.2.2016.